

周报(10.21-10.27)

本周工作:

1. 搭建环境, 开始做 web 端的气象可视化
2. 考虑了一些可能会遇到的技术问题的可能解决方案 (详见下)
3. 初步绘制出了地球的国界线, 但可能存在问题 (详见下)

下周工作:

1. 完成气象可视化 demo 负责的部分

工作记录

1. 需要绘制地图上的国界线, 使用了 `shapefile.js`。根据经验, 可能会遇到类似 `z-fighting` 的问题: 一方面国界线的采样位置与地球表面 `mesh` 不会完全相同, 会存在线的交叉问题 (或者说地图上两点间的直线实际是弧线, 所以直接绘制直线就会与地表交叉而导致无法被观测到); 另一方面存在 `z-fighting`。

考虑的解决方案:

方案 1: 扩大国界线绘制时地球半径

方案 2: 重采样国界线, 然后方案 1

方案 3: `glColorMask(0,0,0,0);glDepthMask(GL_TRUE);` -> 绘制一个较远一点的地球 (或者一个计算好用于遮挡的面) -> `glColorMask(1,1,1,1);glDepthMask(GL_FALSE);` -> 绘制国界线

方案 4: `glDepthRange`

方案 5: 手动剔除反面的线, 剩余线覆盖在最高层

2. WebGL 添加文本的流程:

- 1) 创建一个 `canvas` 并在加入文本: `canvas.getContext('2d').fillText(...)`
- 2) 将 `canvas` 转换为 `texture`: 直接将 `canvas` 传入 `glTexImage2D` 的 `data` 参数即可
- 3) 用 `texture` 进行接下来的操作
- 4) 另一方面, 可以使用 `billboard shader` 使得含有文字纹理的平面永远平行于屏幕; 同时可以使得平面大小与其位置无关。

3. 遥感数据

- 1) 之前出现错误是因为 `endian` 错了, 需要注意。
- 2) 数据为 `[0,300]` 左右范围的 `float`; 目测云的密度与数值大小反相关。
- 3) 16 帧, 有金字塔结构